

PROBABILISTIC MOTION ESTIMATION FOR NEAR REAL-TIME NAVIGATION AND LANDING ON SMALL CELESTIAL BODIES

Cedric Cocaud (1), Takashi Kubota (2)

(1) Dept. of Electrical Engineering, University of Tokyo, ISAS campus 3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, Japan, 81-42-759-8311, cedric.cocaud@ac.jaxa.jp

(2) Institute of Space and Astronautical Science (ISAS-JAXA), ISAS campus 3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, Japan, 81-42-759-8311, kubota@isas.jaxa.jp

Abstract: *In an attempt to avoid the inaccuracies of kinematic models, and the drift error of odometry motion models, this paper investigate the feasibility of using linear visual pose estimation algorithms with Speeded Up Robust Feature descriptors for the motion estimation step of a Particle Filter SLAM. This study fits within the larger scope of investigating the feasibility of Simultaneous Localization and Mapping (SLAM) approaches for real-time spacecraft navigation. While navigation systems aiming at landing on planets such as Mars are incapable of dealing with the additional computational time of SLAM algorithms due to the stringent limitations of onboard space-hardened computers, their low approach speed involved with landing on small celestial bodies such as asteroids or comets may offer viable conditions for the use of SLAM-based navigation systems. With their low computational complexity, linear visual pose estimation algorithms can generate a wide population of pose samples while providing the real-time performance required to ensure proper control of the spacecraft under the stringent computational speed limitation of its onboard computer.*

Keywords: *SLAM, visual navigation, pose estimation, structure from motion.*

1. Introduction

Simultaneous Localization and Mapping (SLAM) has been extensively studied for the past 20 years. The great majority of the work has been applied to rover vehicles moving in 2D and equipped with active sensors such as laser or other range finders, giving unidirectional data on the structure of the environment surrounding the robot [1]. The extension of SLAM to passive sensors such as cameras (referred to as visual-SLAM in the literature) has been of major interest in the past decade, and the integration of this approach to robotic platforms moving in 3D have been the subject of many publications recently (e.g. Unmanned Air Vehicle [2], and Autonomous Underwater Vehicles [3]).

While SLAM is currently being investigated for integration in the next generation of planetary rovers [4], it has infrequently been considered for the navigation and control of spacecraft. One of the main reasons is its additional computational complexity compared to conventional navigation software (i.e. using inertial sensor and extended or simple Kalman filtering – EKF or KF) and the fact that fastest space-hardened onboard computers are currently limited to less than 200 MHz. SLAM approaches are thus considered infeasible for real-time navigation for applications such as landing on Mars or Earth re-entry [5].

On the other hand, the recent Hayabusa mission from JAXA to the asteroid Itokawa has shown that navigation conditions are very different for the Approach Descent and Landing (ADL) phase on small celestial bodies [6]. The very low approach speed in the order of 15 cm/s and the 6 months-long observation period in relative orbit with respect to the celestial body fixed frame allow for more computationally intensive localization and navigation algorithms, update time intervals in the order of minutes still being acceptable during the ADL phase of the mission.

Since current navigation and landing system rely on Kalman Filtering of Inertial Measurement Unit (IMU) data, the authors have proposed a monocular SURF-based visual-SLAM algorithm to cope with the drift error inherent to other dead reckoning techniques, and to make full use of the navigation camera system that is part of the standard sensor array in deep space missions [7]. This approach is based on a Particle Filter (PF) that samples different sets of visual features from a pool of matched features between two subsequent image frames, and generates a series of particles based on the pose estimate calculated for each of those sets, as well as the range of possible scaling factors calculated from scale propagation, or using range sensor data. Due to the computational speed constraint of spacecraft computers, only linear visual pose estimation algorithms can be considered for particle filters if near real-time performance must be met. More robust visual pose estimation techniques such as Bundle Adjustment [8] are simply not fast enough to allow computing more than one pose estimate per SLAM iteration.

This paper focuses on the performance of a linear pose estimation algorithm for generating populations of pose estimates. Their performance is evaluated in the scope of monocular PF-based visual-SLAM for near-real-time spacecraft navigation and control. The performance evaluation parameters considered here are

- the minimum average error achievable
- the average number of particles required to have a minimum number of estimates with error less than a predefined percentage
- the time required to generate this average number of particles

The paper proceeds as follows. Section 2 is a brief overview of the different pose estimation methods used in other SLAM approaches, e.g. Kalman Filter and Particle Filter SLAM. Section 3 described the SLAM approach proposed in this paper, and section 4 discusses the results and the experimental framework that we used to assess the performance of the proposed method.

2. Background

The mathematics behind the pose estimation step of the SLAM requires an appropriate representation of the motion estimate distribution through the parameters of a statistical model (e.g. the average and standard deviation of a Gaussian Model) or a Monte Carlo sampling of this distribution (i.e. Particle Filters) [1].

In theory, maintaining a statistically representative sample or an accurate statistical representation of probable motions ensures the best or most accurate dead reckoning pose estimation, providing that forecasted and observed landmarks are well matched and their position error with respect to the observation probabilistic error model is correctly estimated.

2.1. Extended and standard Kalman Filter-SLAM

While EKF and standard KF-based SLAM avoid the computation of large numbers of probable poses by representing the pose error distribution by a Gaussian error model (or its linearized model for the Extended Kalman Filter), it suffers from the map-scaling problem as the covariance matrix grows quadratically with the number of observations [9]. In order to cope with this, many visual SLAM implementations limit the number of landmarks stored, deleting old ones as new ones are added [10]. This limits how far the robot can go before losing all tracks of the previous places it has visited, increasing the chances of diverging from its real world position and orientation.

2.2. Particle Filter SLAM

The stringent constraint on available computational power onboard spacecraft makes the Particle Filter approach less feasible for real-time performance as the number of pose estimates required to represent the statistical motion distribution increases. This can be referred to as the PF-scaling problem. This is not to be confused with the map-scaling problem which addresses the issues related with handling large database of landmarks in the form of covariance matrices for KF or EKF SLAM, or occupancy grids for PF-SLAM [11][12].

According to previous studies, decreasing the number of particles of a PF-SLAM can significantly decrease its performance up to the point where the SLAM algorithm no longer converges [9]. This is under the assumption that the pose estimation generated at each SLAM iteration are still drawn from a random Monte Carlo process sampling some motion model of the mobile robot.

The divergence of the algorithm observed with fewer particles can come from two factors, the non-homogeneous representation of the possible robot motion distribution, and the premature convergence of the algorithm to a subset of particles that evaluates as more probable in the short-term, but would be less accurate on the long-term given all the future observation to come [13][14].

In practice, the PF-SLAM may still provide accurate motion estimates with fewer particles. This can be done by either using particle resampling techniques that can maintain a representative distribution of the robot poses and landmarks' position [11], or by having pose estimates that are closer to the true motion than the full breadth of all possible poses under some probability distribution. The work of [15] is of particular interest in this area, reporting a technique providing better and fewer estimates by implementing a proposal distribution that considers the accuracy of the robot's sensors, and ensuring particle diversity using an adaptive resampling technique

2.3. The Pose Estimation Step

The problem of finding pose estimates that closely follows the true robot pose is of course the essence of the problem that Particle Filter attempts to solve. The PF algorithm constructs a large number of probable poses called particles, and samples this population according to how well each of them can forecast the position of landmarks being observed at any given time. The main models that have been used to generate population of possible pose estimates are the predictive kinematic model and the odometry motion model [1]. The former takes as input the robot controls (e.g. controlled velocity), the error distribution associated with these controls, the pose estimates sampled from a previous iteration, and output a series of probable poses. The odometry model takes as input the pose estimates sampled from a previous iteration, the measurements of motion sensors (e.g. accelerometers and gyros), and output probable poses based on the error distribution associated with these sensors.

Since both models suffer from drift error and slippage inherent to either the sensor dynamics or the inaccuracies in the mathematical model, a third alternative would be to use visual pose estimation techniques. Several works have dealt with combining motion model and visual-data [2][16], or odometry data [17][18][19].

2.4. Non-linearity of visual pose estimation

KF or EKF-SLAM uses a Gaussian error model for describing the uncertainty over the position of observed landmarks and the pose of the robot. The linearization remains valid when short distances are covered before landmarks are revisited, but might fail when large loops must be closed and the cumulative rotation error introduces unrecoverable non-linearities. Recent works have dealt with this by decoupling the SLAM constraints and using non-linear optimization algorithms

[14][20][21]. The linearization problem is especially important for Visual-SLAM as linearization models fails to capture the true non-Gaussian error distribution of observations acquired from cameras.

In the scope of monocular computer vision systems, observations consist of landmark's 3D position that are triangulated according to 2D-visual feature points matched across two or more images taken at different time. The observation error distribution is a function of the error over the visual feature position in each image frame, the misalignment (or matching error) between matched features, and the triangulation error. While intensity based feature tracking algorithms (e.g. Kanade-Lucas-Tomasi [22]) or other optical-flow based methods may yield an acceptable error range when the process is linearized according to KF-based techniques [23], the matching error distribution may diverge significantly from a locally linearized model when the matching criterion is not related to small position changes. This is the case when using feature descriptors such as SIFT [24], or SURF [25] that are matched based on scale and intensity gradient patterns rather than their relative distance of their image coordinates. In other word, a feature descriptor matching error is chromatically distributed rather than geometrically distributed, implying that small variations in the matching parameters may lead to large variations of feature positions, and in turn, large triangulation errors.

Thus, KF-based techniques are more likely to diverge when using feature descriptors than more robust probabilistic methods which can handle outlying motion estimates by maintaining several estimates at any given time. This is the main motivation for developing the system proposed in this paper. Although some work have been dealing explicitly with the problem of evaluating the uncertainty over triangulated features from different camera pose [26][27], this paper proposes an alternative Monte-Carlo method which does not need to model over feature triangulation as a linear Gaussian distribution.

3. Visual Navigation System

3.1. System overview

The algorithm of the SLAM-based visual navigation system is as follows:

1. The first step consists of acquiring image frames from the navigation cameras and screening them for further processing based on their information content. Information content is evaluated based on a fuzzy-logic histogram analysis measuring the levels of three factors:
 - (i) The amount of dark areas that would lead to unusable visual features
 - (ii) The probability that the image is corrupted
 - (iii) How well the image histogram matches template distribution from other asteroid views

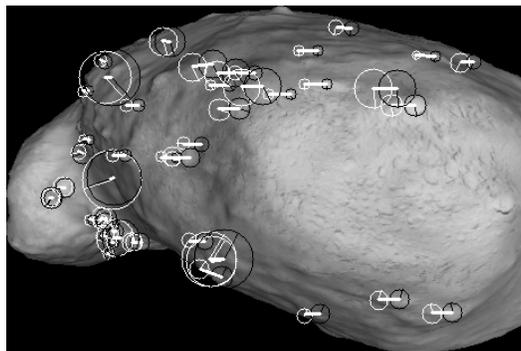


Figure 1. SURF feature extraction and matching (MIGRATE simulated image of Itokawa asteroid)

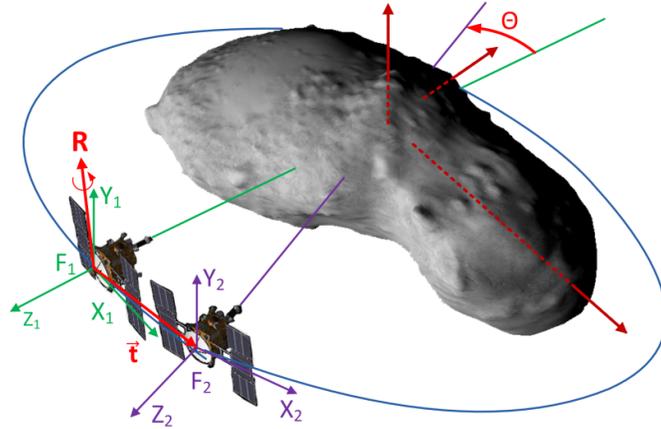


Figure 2. Rigid motion in 3D space of the spacecraft's camera between two successive image frames

2. These three factors are combined through a fuzzy logic module which makes a final pass/no pass decision according to its expert knowledge rule-base.
3. The second step consists of selecting regions of interest in the image, and extracting Speed Up Robust Features (SURFs) [25]. SURF's are robust to illumination changes, orientation, and scale, and can cope with small affine and perspective changes. The feature extraction process is computationally intensive, requiring up to 6 seconds for 100 features on a 512x512 image on a space-hardened computer with a computer speed of 200 MHz. Figure 1 shows the feature extraction process after screening out the dark regions of the image corresponding to empty space.
4. The third step takes care of matching SURF features between two successive images using as metric the Euclidean distance between the features' descriptor (see Figure 1).
5. The fourth step consists in selecting different random subsets of matched features (from the overall population of matched features at a given time) with a fixed size determined by the visual algorithm selected, e.g. 8 matched features for the 8-point algorithm.
6. The fifth, and final step as far as the study presented here is concerned, is to sample scaled camera motions and the associated scaled triangulated feature positions as detailed in section 3.2. The camera calibration matrix and the rigid transform between the camera and the range sensor are assumed to be known and constant.

3.2. Monte-Carlo pose estimation and scaling algorithm

The goal of this algorithm is to generate a representative population of possible scaled camera motions based on a Monte-Carlo sampling of the SURF feature pairs matched in step (4) of the process described in section 3.1. The general concept is to generate a first population of *unscaled* camera motion estimates, and to expand each estimate into a sub-population of *scaled* camera motions based on all the possible scale factors calculated according to range sensor data and previously triangulated features matching the new observations (i.e. the unscaled triangulated features between the pair of image frame from which the unscaled camera motion was calculated).

Inputs:

N_m pairs of matched SURF features positions \mathbf{u}_i in F_1 and \mathbf{u}_i' in F_2 (with $i = 1 \dots m$)

\mathbf{K} , the [3x3] camera calibration matrix

$P_1 = [\mathbf{I} \mid 0]$, the camera matrix for the first camera reference frame F_1 (see Figure 2)

Output:

A population of hypotheses (i.e. particles); the $[i]^{\text{th}}$ particle at time step t_k being described as the scaled camera motion and all N_m triangulated features, conditioned on the N_f image feature points and the k^{th} scale factor.

$$p_t^{[i]} = \left\{ R, T^s, \Theta_{1,t_k}^s \dots \Theta_{j,t_k}^s \dots \Theta_{N_m,t_k}^s \mid \theta_1 \dots \theta_{N_f}, s_l \right\} \quad (1)$$

with $[R, T^s]$, the rotation and translation of the camera between frame F_1 and F_2 (see Figure 2), scaled from $[R, T^u]$ with the l^{th} scale factor s_l drawn from a total number of N_S possible scaling factors,

$[\Theta_{1,t_k}^s \dots \Theta_{N_m,t_k}^s]$, the N_m SURF feature pairs observed at time t_k between F_1 and F_2 , triangulated based on the camera motion $[R, T^s]$,

$[\theta_1 \dots \theta_{N_f}]$, the N_f matched SURF feature pairs used to calculate the unscaled camera motion $[R, T^u]$, with $\theta_i = (\mathbf{u}_i, \mathbf{u}'_i)$

Algorithm:

1. Sample N_P different sets of N_f feature pairs (from a total population of N_m SURF feature pairs)
2. Find the unscaled camera motion for each set p_i of the N_P feature pairs sets:

- 2.1. Compute the normalized image points corresponding to the matched features of the set:

$$\begin{cases} \mathbf{x}_i = \mathbf{K}^{-1} \mathbf{u}_i \\ \mathbf{x}'_i = \mathbf{K}^{-1} \mathbf{u}'_i \end{cases} \quad (2)$$

with \mathbf{u}_i , the i^{th} feature point in image coordinates $[u, v, 1]_i$
 \mathbf{x}_i , the i^{th} feature point in normalized image coordinates $[x, y, z]_i$

- 2.2. Solve for the Essential matrix E describing the camera motion based on the two-points correspondence equation applied to the N_f normalized image point pairs:

$$\mathbf{x}'_i{}^T E \mathbf{x}_i = 0 \quad (3)$$

- 2.3. Based on the basic definition of the Essential matrix,

$$E = [T^u]_{\times} R \quad (4)$$

with R , rotation matrix
 $[T^u]_{\times}$, the skew symmetric matrix of the unscaled translation vector T^u

perform the Singular Value Decomposition (SVD) to retrieve the four possible $[R, T^u]$ configurations of the Essential Matrix [28]:

$$\text{SVD}(E) = U_{(3 \times 3)} \text{diag}_{(3 \times 3)}(r, r, 0) V_{(3 \times 3)}^T \quad (5)$$

$$[R, T^u] = [U W V^T, \pm u_3] \cup [U W^T V^T, \pm u_3] \quad (6)$$

with

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

r , the double singular value of the Essential matrix
 u_3 , the 3rd column of matrix $U_{(3 \times 3)}$

- 2.4. Triangulate one feature pair amongst the N_f pairs of the feature pair set p_i , and find the single camera motion $[R, T^u]$ for which the triangulated 3D feature pair has a positive z-coordinate in both frames F_1 and F_2 (i.e. the point is in front of both camera pose).
- 2.5. Triangulate all SURF feature pairs $[\theta_1 \dots \theta_{N_m}]$ observed at time t_k based on the unscaled motion $[R, T^u]_{p_i}$ and obtain the unscaled triangulated set of features $[\Theta_{1,t_k}^u \dots \Theta_{N_m,t_k}^u]_{p_i}$
3. Find all the possible scaling factors for each estimate p_i of the N_p camera motion estimates:

- 3.1. Calculate the range sensor's target point 3D coordinates X_{R/F_j} with respect to the camera reference frame F_j ($j = 1, 2$) based on the known rigid transform between the range sensor and the camera, as well as the corresponding image point \mathbf{x}_{R/F_j} .

$$\begin{aligned} X_{R/F_j} &= R_{R/F_j} [0, 0, d_R]^T + T_{R/F_j} \\ \mathbf{x}_{R/F_j} &= \frac{1}{z_{R/F_j}} X_{R/F_j} \end{aligned} \quad (7)$$

with $[R_{R/F_j}, T_{R/F_j}]$, the rigid rotation and translation from the range sensor frame to the camera reference frame F_j (the transform is the same for F_1 and F_2)
 d_R , the measured distance from the range sensor

- 3.2. *Scale from range sensor*: Find the nearest feature point \mathbf{x}_i of \mathbf{x}_{R/F_j} within a maximum radius r_{max} in the image plane, and calculate the scale factor s_R by assuming that the depth of the associated points \mathbf{x}_i and X_{R/F_j} are approximately equal

$$X_{R/F_j(Z)} = s_R \cdot \mathbf{x}_{R/F_j(Z)} \quad (8)$$

- 3.3. *Scale propagation*: find all previously triangulated and scaled feature points $\Theta_{n,t < t_k}^s$ that match the visual signature (i.e. SURF feature descriptor) of the currently observed features $[\theta_1 \dots \theta_{N_m}]_{t_k}$ with unscaled triangulated coordinates $[\Theta_{1,t_k}^u \dots \Theta_{N_m,t_k}^u]_{p_i}$, and calculate the scale factor for each of the N_s matches:

$$\Theta_{j,t < t_k}^s = s_n \cdot \Theta_{j,t_k}^u \quad (9)$$

- 3.4. Create $(N_s + 1)$ scaled camera motion estimates from the unscaled estimate p_i , the N_s scale factors s_n from the scale propagation step, and the range sensor scale factor s_R :

$$[R, T^s]_l = [R, (s_l \cdot T^u)] \quad (10)$$

- 3.5. For each scaled camera motion l created, scale all the unscaled feature points:

$$[\Theta_{1,t_k}^s \dots \Theta_{N_m,t_k}^s]_l = s_l \cdot [\Theta_{1,t_k}^u \dots \Theta_{N_m,t_k}^u]_{p_i} \quad (11)$$

4. Results and discussion

4.1. Experimental Framework: the MIGRATE simulator

The Micro Gravity Touchdown and Exploration (MIGRATE) simulator was developed in-house at JAXA's Institute of Space and Astronautical Science. It is a computer graphics 3D engine with an integrated physics engine previously used to test landing scenarios during the Hayabusa mission. It simulates the dynamics of the asteroid Itokawa and the spacecraft Hayabusa, as well as the LIDAR and IMU sensors.

Camera images are generated in real-time and processed by the navigation and control system which is implemented as a completely independent module for portability to other experimental platform. The computational speed of the control system is scaled down based on the maximum theoretical speed of currently available space-hardened computers, i.e. 200 MHz.

4.2. Results

The simulation for which the following results are shown sets the spacecraft facing the asteroid at a distance of 500 meters, staying stationary for an entire revolution of the asteroid. The spacecraft is thus set in an observation mode, and it attempts to track its relative pose change between successive pair of images by tracking the visual features on the ground. The simulation time is scaled according to the rotation speed of the asteroid (taken to be the same as Itokawa's period of rotation), and the onboard simulated processing speed set here to 200 MHz.

The following preliminary results aim to assess the feasibility of the proposed approach in terms of accuracy and time performance. More specifically, the two questions addressed are the following: (i) what is the maximum accuracy of the overall population of camera motion estimates (ii) can the number of camera motion estimates required to have a minimum desired accuracy (i.e. a given maximum error) be processed in near real-time.

The answer to the first question is given in Figure 3. Since the accuracy of each camera motion estimate will be different from one another, the two graphs show instead the percentage of camera motion estimates with maximum bounded error of 20, 10 and 5%, over the entire population of estimates at each time step.

As shown in Figure 3, more than 95 % of the camera rotation estimates have an error less than 5% with respect to the true camera motion (the latter being known exactly as the spacecraft position is fully determined by the MIGRATE simulator at each time step).

On the other hand, we can see that the bottleneck for accuracy is the translation. The average percentage of motion estimates with a translation error of less than 20% corresponds to only 8% of the estimate population. The percentage of estimates drops to an average of 1 % for translation error of less than 5%.

The error for both rotation and translation are taken here as the angle between the true pose of the camera and the estimated one, divided by π (the maximum angular deviation being 180 degrees according to this metric).

Although it seems that very few motion estimates have a high accuracy for the combined rotation and translation components, the results still show that enough reliable estimates can be generated if the total population size (i.e. the number of camera motion estimate) is set to a sufficiently high value, 300 estimates being enough here to have a few ones with error less than 5%.

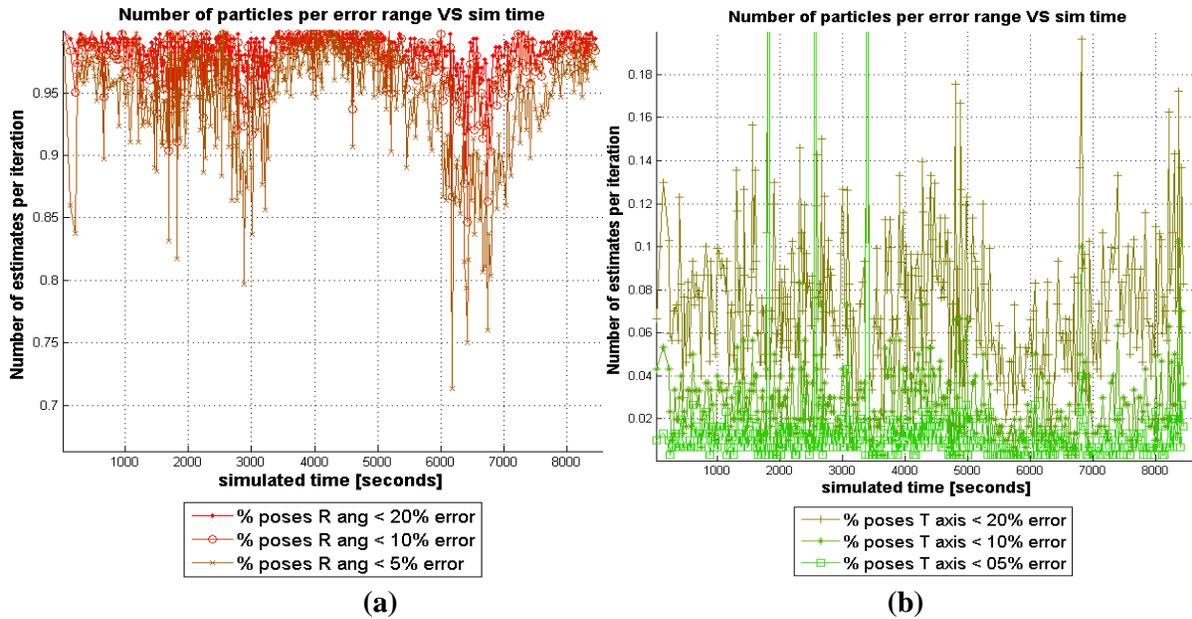


Figure 3: Accuracy of the population of relative camera motion estimates for one complete revolution around the asteroid (simulated time scaled for a 200 MHz computer) (a) rotation estimate (b) Translation direction estimate

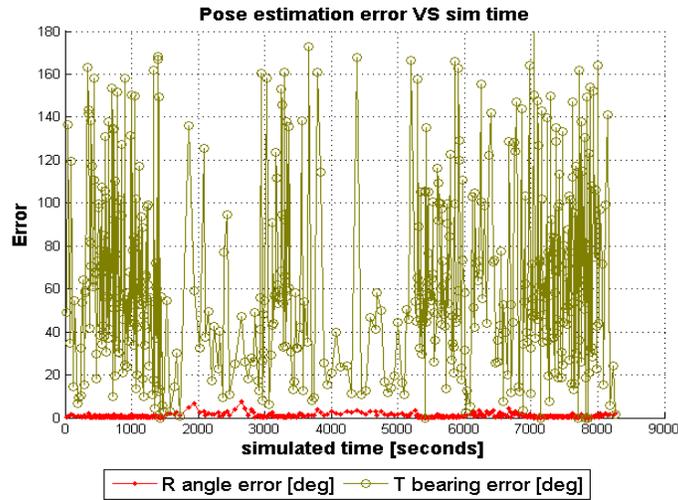


Figure 4: Accuracy of a single camera motion estimate with the conventional RANSAC method using the same linear 7-point pose estimation algorithm for one complete revolution around the asteroid (simulated time scaled for a 200 MHz computer)

In order to compare the performance of the proposed algorithm with a conventional method with comparable time performance and using the same linear visual pose estimation algorithm (i.e. the 7-point algorithm), the camera motion estimation step has been performed under the same simulated conditions using the well known RANSAC algorithm. As shown in Figure 4, the conventional approach shows a relatively high degree of accuracy with respect to the rotation estimates with a maximum error of 10 degrees. However, the translation estimates show error rising up to 160 degrees (corresponding to a percentage error of 88 % according to the metric described above). The conventional approach giving only a single estimate at each time step, we can see that the spacecraft's pose estimate would quickly diverge during the observation phase.

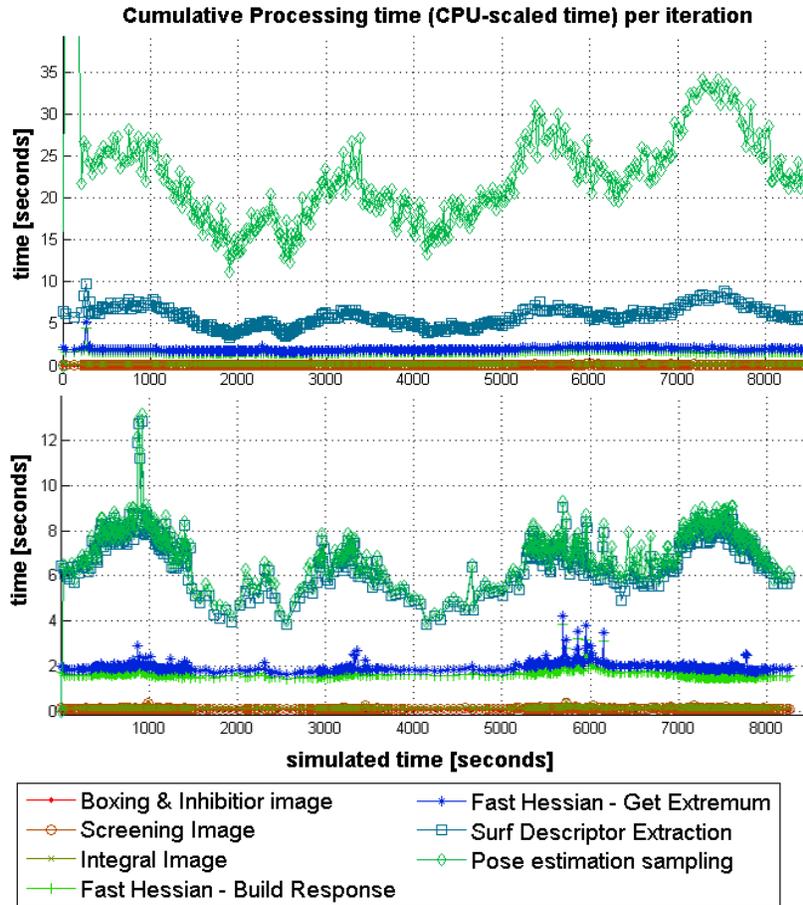


Figure 5: Time response (all time scaled for a 200 MHz computer) of (a. above) the probabilistic algorithm for a population of 300 camera motion estimates (b. below) RANSAC-7 pt algorithm for the computation of a single camera motion estimate

Addressing the second question regarding the feasibility of generating in near-real-time a sufficiently large population of motion estimates, we can see in Figure 5 (a) that the generation of 300 motion estimate can actually be achieved in an average of 21.77 seconds. Taking an average number of motion estimates of 4% for localization errors of less than 10%, we end up with at least 12 motion estimates which can eventually be selected by the SLAM during the hypothesis selection step of the algorithm. A sufficient number of accurate camera motion estimates may thus be generated in near-real-time, as long as the selection phase of the SLAM is able to identify these estimates as more likely than the others.

The time performance of the conventional RANSAC approach is shown in Figure 5 (b). Its average time response is 7.23 seconds. Although the conventional approach is almost three times faster than the proposed approach, the proposed approach is still operable in near-real-time, and offers motion estimates with much greater accuracies.

Based on the preliminary results shown above, it can already be said that the proposed approach can yield results with much greater accuracy than conventional ones such as RANSAC, given the conditions and the type of features with which the navigation system must deal.

Although the number of camera motion estimates that need to be generated slows the algorithm by a factor of 3 compared to the conventional RANSAC method, it remains operable in near-real-time.

Moreover, the percentage of high-accuracy estimates could be further increased at each time step while maintaining near-real-time capabilities by replacing the 7-point algorithm with a more accurate one, such as the 5-point linear visual pose estimation algorithm.

5. Conclusion

Addressing the need for robust pinpoint landing capabilities for small celestial body missions, this research proposes a robust probabilistic monocular navigation scheme based on the Simultaneous Localization and Mapping (SLAM) approach. This online navigation scheme provides attitude and position (or pose) estimates during the Approach, Descent and Landing phase, while simultaneously mapping the topography of the celestial body, and is meant to offer a solution to the navigation shortcomings of the Japanese mission Hayabusa, and to extend the capabilities of the spacecraft navigation system of the missions currently planned by JAXA, namely Hayabusa II and Hayabusa Mark II.

This approach relies on combining data from a camera and a range sensor (e.g. LIDAR) in order to maintain several hypotheses of the most likely spacecraft pose and landmark position at any given time. In contrast with most previous works which used KF-based techniques to represent the scaling uncertainty over the landmarks and camera motion, the proposed scheme uses a two-stage Monte-Carlo method to represent the population of all possible spacecraft motion between successive camera image pairs, and to represent the population of all possible scaling factors in order to express each motion in world-scaled coordinates.

The Monte-Carlo method draws candidate camera motions by sub sampling the population of visual feature pairs, and using a linear visual pose estimation algorithm on these feature subsets. Although less accurate than non-linear optimization algorithms, results have shown that the use of a linear algorithm yields accurate estimates on a probabilistic basis, while allowing the visual navigation system to operate in near-real-time on currently available space-hardened computer (i.e. with a CPU speed less than 200 MHz).

6. References

- [1] Thrun S., Burgard W., and Fox D., “Probabilistic Robotics”, MIT Press, 2005
- [2] Artieda J., Sebastian J.M., Campoy P., Corraera J.F., Mondragón I.F., Martínez C., Olivares M., "Visual 3-D SLAM from UAVs", Jour. of Intelligent and Robotic Systems, Vol 55, pp. 299-321, 2009
- [3] West M.E., Syrmos V.L., “Navigation of an Autonomous Underwater Vehicle(AUV) Using Robust SLAM”, IEEE Proc. of the Int. Conf. on Control Applications (CCA), Munich, Germany, 4-6 Oct., pp. 1801-1806, 2006
- [4] Bakambu J.N., Langley C., Pushpanathan G., MacLean W.J., Mukherji R., Dupuis E., “Planetary Rover Visual Motion Estimation improvement for Autonomous, Intelligent, and Robust Guidance, Navigation and Control”, Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Sapporo, Japan, 30 Aug.-1 Sept., pp. 552-559, 2010
- [5] Trawny N., Mourikis A.I., Roumeliotis S.I., Johnson A.E., Montgomery J.F., “Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks”, Jour. of Field Robotics, Vol 24 (5), pp. 357–378, 2007

- [6] Kubota T., Hashimoto T., Sawai S., Kawaguchi J., Ninomiya K., Uo M., Baba K., "An autonomous navigation and guidance system for MUSES-C asteroid landing", *Acta Astronautica*, Vol 52, pp. 125-131, 2003
- [7] Cocaud C., Kubota T., "SURF-Based SLAM Scheme using Octree Occupancy Grid for Autonomous Landing on Asteroids", *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Sapporo, Japan, 30 Aug.-1 Sept., pp. 275-282, 2010
- [8] Triggs B., McLauchlan P., Hartley R., Fitzgibbon A., "Bundle Adjustment - A Modern Synthesis", *Proc. of the International Workshop on Vision Algorithms (ICCV)*, Springer-Verlag, pp. 298-372, 1999
- [9] Montemerlo M., "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association", Ph.D. thesis, Carnegie Mellon University, United States, 2003
- [10] Davidson A.J., "Real-Time Simultaneous Localisation and Mapping with a Single Camera", *IEEE Proc. 9th Int. Conf. on Computer Vision (ICCV)*, Vol 2, Nice, France, 13-16 Oct., pp. 1403-1410, 2003
- [11] Crisetti G., Stachniss C., Burgard W., "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters", *IEEE Transactions on Robotics*, Vol 23 (1), Feb., pp. 34-36, 2007
- [12] Fairfield N., Kantor G.A., and Wettergreen D., "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels", *Jour. of Field Robotics*, Vol 24 (1/2), pp. 3-21, 2007
- [13] Begum M., Mann G.K.I., Gosine R.G., "Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots", *Applied Soft Computing*, Vol 8, pp. 150-165, 2008
- [14] Olson E., Leonard J., Teller S., "Fast Iterative Alignment of Pose Graphs with Poor Initial Estimates", *IEEE Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida, 15-19 May, pp. 2262-2269, 2006
- [15] Grisetti G., Stachniss C., Burgard W., "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters", *IEEE Trans. on Robotics*, Vol 23 (1), Feb., pp. 34-36, 2007
- [16] Munguia R., Grau A., "Monocular SLAM for Visual Odometry", *IEEE Int. Symp. on Intelligent Signal Processing (WISP)*, Madrid, Spain, 3-5 Oct., pp. 1-6, 2007
- [17] Bleser G., Stricker D., "Using the Marginalised Particle Filter for Real-Time Visual-Inertial Sensor Fusion", *IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, Cambridge, UK, 15-18 Sept., pp. 15-18, 2008
- [18] Persson M., Duckett T., Lilienthal A.J., "Fusion of aerial images and sensor data from a ground vehicle for improved semantic mapping", *Jour. of Robotics and Autonomous Systems*, Vol 56, pp. 483-492, 2008
- [19] Kima J., Sukkarihb S., "Real-time implementation of airborne inertial-SLAM", *Jour. of Robotics and Autonomous Systems*, Vol 55, pp. 62-71, 2007
- [20] Grisetti G., Stachniss C., Grzonka S., Burgard W., "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent", *Proc. of Robotics: science and systems*, 2007

- [21] Estrada C., Neira J., Tardós J.D., "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments" IEEE Trans. on Robotics, Vol 21 (4), Aug., pp. 588-596, 2005
- [22] Shi J., Tomasi C., "Good Features to Track", IEEE Proc. Conf. on Computer Vision and Pattern Recognition (CVPR), Seattle, USA, 13-15 June, pp. 593-600, 1994
- [23] Kendoula F., Fantoni I., Nonamib K., "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles", Jour. of Robotics and Autonomous Systems, Vol 57, pp. 591-602, 2009
- [24] Lowe, David G. "Object recognition from local scale-invariant features". IEEE Proc. of the Int. Conf. on Computer Vision (ICCV), Vol 2, Kerkyra, Greece, 20-25 Sept., pp. 1150-1157, 1999
- [25] Bay H., Ess A., Tuytelaars T., Van Gool L., "SURF: Speeded Up Robust Features", Proc. Conf. on Computer Vision and Image Understanding (CVIU), Vol 110 (3), pp. 346-359, 2008
- [26] Eade E., Drummond T., "Scalable Monocular SLAM", IEEE Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR), New York, USA, 17-22 June, pp. 469-476, 2006
- [27] Marzorati D., Matteucci M., Migliore D., Sorrenti D. G., "Monocular SLAM with Inverse Scaling Parametrization", Proc. of the British Machine Vision Conference, Vol 12 (44), 2008
- [28] Hartley R., Zisserman A., "Multiple View Geometry in Computer Vision Second Edition", Cambridge University Press, 2nd Edition, 2003